

Vytěžování dat, cvičení 1: Úvod do Matlabu

Michael Anděl, Miroslav Čepek

23. 9. 2014



Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Fakulta elektrotechnická, ČVUT

Jiří Kléma

email : klema@fel.cvut.cz

Filip Železný

Místnost: KN-E 432

email : zelezny@fel.cvut.cz

Místnost: KN-E 201

Michael Anděl

email : andelmi2@fel.cvut.cz

Miroslav Čepěk

Místnost: KN-E 224

email : cepekmir@fel.cvut.cz

Konzultace jsou možné po předchozí domluvě. Preferovaná forma konzultací je přes diskusní fórum předmětu na <https://cw.felk.cvut.cz/forum/forum-164.html>.

Úvod do předmětu

Proč Matlab?

Programování pro Matlab

Zápočtová úloha

Byli bychom rádi, kdybyste si z předmětu odnesli

- ▶ jaké jsou základní úlohy ve vytěžování dat (tím nemyslíme tento předmět),
- ▶ jak základní metody pro tvorbu modelů,
- ▶ představu o vyhodnocování přesnosti a úspěšnosti modelů,
- ▶ širší povědomí o data miningu.

- ▶ Stránky předmětu: <https://cw.felk.cvut.cz/doku.php/courses/a7b36vyd/start>
- ▶ Náplň cvičení bude odevzdání domácí úlohy a konzultace k zadané domácí úloze. Tj. jejich obsah budou tvořit především dotazy na probranou látku na přednáškách a zadané úlohy.
- ▶ Na začátku každého cvičení bude zadána jedna zápočtová úloha týkající se látky z poslední přednášky.
- ▶ Z každé úlohy se vypracovává krátký protokol, rozsahem přibližně jedna strana A4.
- ▶ Celkem bude 11 zápočtových úloh a všechny je musíte odevzdat. Za každou úlohu můžete získat až 5 bodů.
- ▶ Dohromady musíte za úlohy získat alespoň 30 bodů.

- ▶ Každý domácí úkol (kromě prvních dvou) bude na procvičení/vyzkoušení látky probírané na přednášce \Rightarrow chodte na přednášky!
- ▶ K získání zápočtu musíte odevzdat všech 11 domácích úkolů.
 - ▶ Na vypracování každého úkolu bude jeden týden, pokud nebude řečeno jinak.
 - ▶ Zprávu budete odevzdávat přes upload systém na stránkách předmětu.
 - ▶ Deadline na odevzdání je půlnoc z pondělí na úterý daného týdne.
- ▶ V případě **závažných** důvodů může cvičící povolit pozdní odevzdání. Problémy řešte co nejdříve!! (A pokud možno, dopředu).

- ▶ Zpráva musí být v upload systému na stránkách předmětu do zadaného deadlinu.
- ▶ Pokud má úloha i programovací část, můžete ji ladit ještě i po deadlinu až do odevzdání na cvičení.
- ▶ Stejně tak, pokud při odevzdání na cvičení vyplují na povrch drobné nedostatky, máte šanci je opravit přímo na cvičení bez ztráty bodů.
- ▶ Body se udělují stylem SPLNĚNO/NESPLNĚNO.
 - ▶ Splnění a odevzdání včas – 5 bodů.
 - ▶ Splnění a odevzdání s týdením zpožděním – 2 body.
 - ▶ Splnění a odevzdání s dvoutýdenním zpožděním – 0 body (ale stále nárok na zápočet).
 - ▶ Později – neudělení zápočtu.

- ▶ Body ze cvičení si ponesete ke zkoušce, kde vám budou k užítku :).
- ▶ Zkouška bude hlavně písemná – z písemky můžete získat až 45 bodů.

Výsledná známka bude dána součtem bodů ze zkouškové písemky a cvičení:

ECTS známka	A	B	C	D	E	F
Počet bodů	100-90	89-80	79-70	69-60	59-50	50-0

- ▶ Matlab je SW pro vědecko-technické výpočty a de facto průmyslový standard v mnoha odvětvích.
- ▶ Vhodný pro rychlé prototypování a zkoušení aplikací.
- ▶ Obsahuje skriptovací jazyk se spoustou knihoven pro různé oblasti (včetně vytěžování dat).
- ▶ Matlab je v tomto předmětu kompromis mezi "klikacími" nástroji typu RapidMiner a programováním v jazycích typu Java.
- ▶ Dovolí nám hrát si s implementací algoritmů, ale spoustu věcí řeší interně za vás.

- ▶ FEL má multilicenci, kterou můžete využít.
- ▶ Kopii instalačního DVD můžete získat na <http://www.fel.cvut.cz/user-info/matlab.html>
- ▶ Zde se musíte přihlásit hlavním přístupovým heslem a pak už můžete stahovat a instalovat.

Základní uživatelské rozhraní

The screenshot displays the MATLAB 7.11.0 (R2010b) environment. The **Current Folder** panel at the top shows the path `/Users/cepekml/Documents/MATLAB`, which is highlighted with a red box and labeled "Aktuální pracovní adresář". The **Command Window** shows the following code and output:

```
>> x = [1 1 2 2 4 5 6];  
>> mode(x)  
  
ans =  
  
1  
  
>> x = [1 1 2 2 4 5 6 6 6];  
>> mode(x)  
  
ans =  
  
6
```

Below the code, the help text for the `mode` function is visible. A red annotation "Příkazový řádek pro zadávání příkazů a jejich výstup" points to the command input area. The **Workspace** panel shows a table of variables:

Name	Value	Min	Max
A	<10x5 double>	0.01...	10
B	[1,2,3,4,5,6]	1	6
C	[1,2,3,10,11,12]	7	12
ans	[1,2,3,4,5,6,6,6]	1	6
i	[1,2,3,4,5]	1	4
row	[1,2,3,4,5]	1	3
sl	[1,2,3,4,5]	1	3
str	[1,2,3,4,5]	1	3
x	[1,1,2,2,4,5,6,6,6]	1	6
y	[2,4]	2	4

A red annotation "Aktuálně definované proměnné a jejich hodnoty (zkuste dvojklik)" points to this table. The **Command History** panel shows the following commands:

```
A(1:10) > 0.5);  
fprintf('%d', 5)  
fprintf('%d\n', 5)  
disp(5)  
disp(x)  
for i = 1:6 10000 4 3.1415 2.7 1.41  
disp(i); end  
factorial(5)  
help  
A > 0.5  
help mode  
x = [1 1 2 2 4 5 6];  
mode(x)  
x = [1 1 2 2 4 5 6 6 6];  
mode(x)
```

A red annotation "Historie příkazů" points to the Command History panel. The **Current Folder** panel shows a file named `factorial.m` with a red annotation "Obsah aktuálního adresáře" pointing to it.

Do příkazového okna zkuste zadat následující příkazy. Jaký je jejich význam a co je výsledkem?

- ▶ $1 + 5$
- ▶ Ukládá se někam výsledek?
- ▶ $x = 3$
- ▶ $y = 1 * 6$
- ▶ $x + y$
- ▶ $z = x + y$
- ▶ $z = x + y$;
- ▶ Jaký je rozdíl mezi předchozími dvěma příkazy?

- ▶ Hlavní síla Matlabu spočívá v práci s maticemi.
- ▶ Jak vypadají matice z následujících příkazů?

```
A = [1 2; 3 4; 5 6]
```

```
B = [1 2 3; 4 5 6]
```

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

- ▶ Jaké znáte maticové operace?
- ▶ Maticové sčítání, odčítání, násobení, dělení, transpozice
- ▶ $C = [7 \ 8; \ 9 \ 10; \ 11 \ 12]$
- ▶ $A + C$
- ▶ $A * B$
- ▶ $A .* C$
- ▶ Jak se liší výsledky posledních dvou příkazů?
- ▶ Co dělá A' ?

- ▶ Matice náhodných čísel `rand(<počet prvků v 1. dimenzi>, <v 2. dimenzi>, ...)`
 - ▶ například `A = rand(10, 5)`
- ▶ Velikost matice, počet prvků v jednotlivých dimenzích – `size(A)`
- ▶ Matice samých nul – `zeros(10, 6)`
- ▶ Inverzní matice – `inv(A)`
- ▶ Vlastní čísla – `eig(A)`
- ▶ Determinant – `det(A)`
- ▶ Indikace splnění podmínky – `A > 0.5`
- ▶ Vybere indexy z matice na základě podmínky – `find(A > 0.5)`

Existuje několik možností získání nápovědy pro příkaz Matlabu.

- ▶ Jednoduchá textová nápověda – příkaz `help <příkaz>`
- ▶ Hypertextová nápověda – příkaz `doc <příkaz>`
- ▶ Mathworld Knowledge Base –
<http://www.mathworks.com/help/techdoc/>

- ▶ Vektory jsou matice, které mají jen jeden řádek/sloupec.
- ▶ Vytvořte řádkový a sloupcový vektor hodnot 1, 2, 3, 4, 5, 6.
- ▶ `row = [1 2 3 4 5 6]`
- ▶ `s1 = [1; 2; 3; 4; 5; 6]`
- ▶ Jak vytvořím z řádkového vektoru sloupcový?
- ▶ Transpozicí – `row'`

Pokud potřebujete vytvořit posloupnost čísel, můžete využít příslušný operátor ":" (dvojtečka).

Pomocí něj vygeneruje vektor, který obsahuje posloupnost čísel v zadaném rozmezí – zkuste

- ▶ `[1:10]`; `[-5:5]`; `[5.4:15.6]`
- ▶ můžete zadat i krok, se kterým se posloupnost mění. Zkuste: `[1:0.5:10]`; `[5:-1:5]`; `[5.4:0.2:15.6]`

Zopakujte $A = \text{rand}(10, 5)$, ať máme všichni stejné rozměry matice.

- ▶ Přístup k jednomu konkrétnímu prvku – $A(1,2)$
- ▶ **POZOR** – indexy se číslují od **1**!
- ▶ Přístup k podmaticím – uvedu rozsahy indexů, které chci v mít podmatici $A(3:5, 1:3)$
- ▶ Místo čísel můžu uvést i vektory indexů, které chci zobrazit.
- ▶ Zkuste zobrazit prvky ve řádcích 1., 5., 3., 8. a sloupcích 3., 2., 1.
- ▶ $x = [1\ 5\ 3\ 8]$; $y = [3\ 2\ 1]$; $A(x,y)$

- ▶ Úplně stejně jako výběr prvků na minulém slajdu, jen výběr umístím na levou stranu přiřazení.
- ▶ Přiřazení jedné hodnoty – přiřadte 10 do levého horního rohu matice.
- ▶ $A(1,1) = 10$
- ▶ Přiřazení do podmatice – přiřadte hodnoty 1, 2, 3, 4 na souřadnice (2,2), (2,4), (4,2), (4,4)
- ▶ $x=[2\ 4]$; $y=[2\ 4]$; $A(x,y) = [1\ 3; 2\ 4]$

- ▶ V Matlabu, stejně jako v jiných skriptovacích jazycích, proměnné nemají pevný datový typ.
- ▶ Základní datové typy jsou:
 - ▶ Čísla, Logické hodnoty, Řetězce
 - ▶ Matice – matice hodnot jednoho datového typu
 - ▶ Struktury – skupina několika pojmenovaných hodnot zabalených do jedné proměnné
 - ▶ Buňková pole (Cell arrays) – pole hodnotu různých datových typů
 - ▶ Odkazy (Handles)
 - ▶ Objekty

Více o datových typech se lze dozvědět na http://www.mathworks.com/help/techdoc/matlab_prog/f2-43934.html

Doteď jsme zkoušeli interaktivní práci se systémem Matlab. Teď zkusíme programování.

Programy se zadávají do tzv. M-souborů (M-file) což je jen textový soubor s koncovkou `.m`. Existují dva typy M-souborů

- ▶ Skripty – obsahují jen posloupnost příkazů Matlabu,
- ▶ Soubory definujících funkci – obsahuje definici funkce, kterou lze využít při interaktivní práci s Matlabem nebo v jiných M-souborech.

Pro editaci obou typů M-souborů můžete použít:

- ▶ libovolný textový editor,
- ▶ editor integrovaný přímo do Matlabu.

My použijeme druhou možnost. Interní editor se spouští buď příkazem `edit` nebo z menu `File > New Script` resp. `File > New Function`.

Základní struktura funkce vypadá takto:

```
function [soucet,rozdil] = SlozitaFunkce(a,b)
%
% Funkce se jmenuje SlozitaFunkce a ma dva vstupni
% parametry - a, b. A dva vystupni parametry -
% soucet, rozdil.
% Tohle je help k funkci SlozitaFunkce. Vypisuje se
% zadanim prikazu 'help SlozitaFunkce'.
%
%Naplneni vystupniho parametru se deje prirazanim
    soucet = a+b;
    rozdil = a-b;
end % Nepovinne end
```

Funkce **musí** být uložena v souboru slozitafunkce.m.

- ▶ Funkci můžete zavolat pomocí jejího jména.
- ▶ `[s, r] = slozitafunkce(10, 4)`
- ▶ Jak Matlab zjistí, kterou funkci voláte?
 - ▶ Matlab se podívá do aktuálního adresáře, zda v aktuálním adresáři existuje soubor `slozitafunkce.m`
 - ▶ Pokud Matlab funkci nenajde v aktuálním adresáři, podívá se na vnitřní proměnné `PATH` na seznam adresářů, které se mají prohledávat a zkusí, zda některý z nich funkci neobsahuje.
- ▶ Z toho plyne, že je důležité, ve kterém adresáři se nacházíte :).

Provádění funkce můžete ovlivnit pomocí standardních konstrukcí, které znáte z jiných programovacích jazyků.

- ▶ Podmínky
 - ▶ if – then – else
 - ▶ switch – case
- ▶ Cykly
 - ▶ for
 - ▶ while

if – then – else

```
if x > 11
    disp('x je vetsi nez 11.');
```

elseif x < 5

```
    disp('x je mensi nez 11 a take mensi nez 5.')
```

else

```
    disp('Je to nejak uplne jinak.');
```

end

switch – case

```
switch x
    case {0, 1, 2, 3}
        disp('x je v intervalu 0-3.');
```

case 4

```
        disp('x je 4.');
```

otherwise

```
        disp('x je uplne jine.')
```

end

For cyklus funguje trochu jinak než jste zvyklí z Javy. For cyklus v Matlabu iteruje přes všechny hodnoty vektoru.

```
for i = 1:10
    fprintf('%d ',i); %muzete pouzit take disp(i)
end
```

```
for i = [1 6 10000 4 3.1415 2.7 1.41]
    disp(i);
end
```

POZOR – použití cyklů je extrémně nevýhodné, protože Matlab parsuje každý řádek znovu \Rightarrow pokud můžete zkuste cykly obejít (například vektorovými nebo maticovými operacemi).

While cyklus je mnohem blíž tomu, co znáte z jiných jazyků.

```
i = 1;
while i < 10
    fprintf('%d',i);
    i = i+1;
end
```

Existují i příkazy na přerušení cyklů

- ▶ break – ukončuje cyklus if $i = 5$, break, end
- ▶ continue – přeskakuje zbytek iterace if $i = 5$, continue, end

```
function fac = factorial(n)
if n < 0
    disp('n musi byt vetsi nez 0!');
    fac = 0;
    return;
end
if n == 0
    fac = 1;
    return;
end
fac = 1;
for i = 1:n
    fac = fac * i;
end
end
```

1. zápočtová úloha (I)

Termín odevzdání: 26.9. 2011 nejpozději v 23:59:59 (SELČ). Do upload systému na stránkách předmětu.

Zadání – vytvořte v Matlabu:

- ▶ funkci `fib_arr(array)`, která pro všechny prvky ve vektoru `array` spočítá fibonacciho hodnotu a vrátí pole těchto hodnot.
- ▶ funkci `stat_props(filename)`, která načte zadaný soubor ve formátu CSV a pro každý **řádek** hodnot vypíše maximum, minimum, medián, průměr, rozptyl, stření hodnotu.

Použití všech funkcí Matlabu je samozřejmě dovoleno!

1. zápočtová úloha (II)

Protokol z první úlohy, který nahrajete na web, bude PDF s textovou zprávou. Bude obsahovat:

- ▶ stručný popis vašeho řešení a okomentované nejpodstatnější části kódu vašich funkcí,
- ▶ **krátká** ukázková data a výsledky,
- ▶ **jednovětný** popis termínů uvedených v zadání (fibonacciho posloupnot, medián, rozptyl, ...).

Funkčnost vašeho kódu předvedete na cvičení.

Poznámka: Pro načítání souboru ve formátu CSV můžete použít funkci `csvread`.

Zde jsou odkazy na další výukové materiály o Matlabu

- ▶ <http://labe.felk.cvut.cz/~posik/y33aui/uvod-do-matlabu/>
- ▶ http://www.mathworks.com/help/techdoc/matlab_product_page.html
- ▶ <http://www.mathworks.com/moler/intro.pdf>
- ▶ <http://www.maths.dundee.ac.uk/~ftp/na-reports/MatlabNotes.pdf>
- ▶ a mnoho dalších...